

Shared Resource Control between Human and Computer

James Hendler*

Reid Wilson

Computer Science Dept.

University of Maryland, College Park

February 6, 1989

1 Introduction

One area of Artificial Intelligence systems for space applications is to serve as a mechanism for telerobotic control. When interaction is called for, it is usually viewed as a passing of control with *either* an AI system *or* a human in complete command of the teleoperated system. Unfortunately, this can cause serious inefficiencies in the planning process, both for the AI system and the human, when the plans and actions of each may cause potential conflicts. If the human were to, for example, remove a bolt which the AI system had previously placed, then it is important for the robot to take this information into account in planning a later action concerning that bolt or its assembly. If this information must be rediscovered by the AI system upon regaining control, the system must spend much time both checking for states already achieved and for replanning the accomplishment of previously achieved actions.

This paper describes the advantages to an AI system of actively monitoring a human's control of a shared resource (such as a telerobotic manipulator). We describe a system in which a simple AI planning program gains efficiency by monitoring a human's actions and recognizing when they cause a change in the system's assumed state of the world. This enables the planner to recognize when an interaction occurs between the human's actions and the system's goals, and allows it to maintain an up-to-date knowledge of the state of the world and thus to (i) inform the human when an action would undo a goal achieved by the system, (ii) inform the human when an action would render a system goal unachievable, and (iii) efficiently replan the establishment of goals after human intervention.

*Funding for this work was provided in part by the Office of Naval Research, the UM Institute for Advanced Computer Studies, and the Systems Research Center

2 Shared Resource Control

The current view of shared resource control between a human and an AI system generally centers on one or the other partner being in full control of the resource at any given time. Thus, in a telerobotic system, either a human operator or an AI system may be controlling an arm. When the human is in control the AI system is quiescent. However, when the AI system is in control, the human is viewed as being in a supervisory role. This allows a human to monitor the system and make sure that no unanticipated problems are caused by the system. (This is frequently the case in telerobotic systems because of the need for a human operator to deal with unanticipated or emergency situations outside of the AI system's problem solving capabilities. This is particularly important when the situation is somewhat fluid, that is, when an earlier unexpected event may cause repercussions not anticipated at system design time. As an example of such an event, consider an undeployed solar collector which might cause power consumption considerations not anticipated at system design time).

It is our contention that in a similar manner to a human's supervision of the AI system, the AI system must monitor the human's use of the shared resource. While we believe that it is important that the human maintains control (i.e. has the final word), the AI component can both gain efficiency and provide some measure of error checking by performing this monitoring. Allowing the AI system to monitor human performance avoids serious inefficiencies in the planning process and provides benefits not possible where the human is not monitored.

Probably the most important benefit to the AI system is its ability to update its internal view of the world to reflect any changes observed during the times of human control. This is very important, because by monitoring, the system will always have an accurate view of the world. Although there is some overhead in such a process, the planner will have to update its state of the world in response to each human action, the benefit provided is that the AI system never needs to reestablish the current state of the world.

The ability of the system to maintain a complete and accurate world model is quite important. In current systems, upon completion of the human's control of the resource, the AI system is left with an incomplete, and possibly incorrect world model. Thus, for example, if a human has moved some object (say a bolt), the robot must either explicitly check that the object is still where it was previous to human intervention (requiring a move to the location of the bolt), or it must assume no harmful interactions will have been caused by the human and try to work with an incomplete state model. Unfortunately, working with an incomplete state model is a major problem for current AI system (a long discussion and literature review of this issue is presented in Sanborn and Hendler, 1988). Thus, the planner is faced by a choice between reestablishing all conditions, which is a highly resource intensive and inefficient process, or it must work from incomplete knowledge with all the attendant dangers. By monitoring the human's actions, to maintain an awareness of the actual state of the world, the planner need not have to rediscover information each time control is returned to it without having to sacrifice information certainty.

Another reason for the need for the AI system monitoring the human is that when partners share control of a resource they may have conflicting, complementary, and/or unrelated goals. In

each case, having the passive agent monitor the actions of the controlling partner allows significant benefits:

- Where a conflict arises, the partners may resolve it early.
- Where cooperation may occur, similar preconditions are not reestablished needlessly.
- Where goals are unrelated, accidental interactions (such as moving a tool required later) are largely alleviated.

In the remainder of this paper we describe a small system which was designed solely to show the efficacy of such monitoring. In the next section we describe the system and show, by way of an example, how the monitoring provides for the sorts of interactions described above. This system is quite limited, and is in no way proposed as an actual telerobotic planning system. Some steps towards that direction are discussed in section 4.

3 An Example System

To demonstrate how shared control better enables the partners to handle interacting goals, we have developed a very simple blocks world system in which control of a “robot” is shared between a human operator and an Artificial Intelligence Planning system. (In section 4 we will discuss limitations of this sort of system viz. realistic telerobotic control.) Instead of having one or the other partner control the robot independently (i.e., the other not receiving simultaneous feedback), both parties are active in monitoring the actions of the other and the effects on their goals. Thus this shared control allows the passive agent will determine how the actions of the controlling agent affect its goals, and react appropriately (interacting as necessary).

In our system the “robot” can be used for the manipulation of items in a simple domain: a set of various blocks of different size and orientations. Neither the blocks nor the robot may be moved off the edge a simple “world” consisting of a simple grid. Only one agent (human or computer) may actively control the robot at one time, although the other receives the commands given to the robot before they are executed. While the AI system is in control, the human is allowed the human to assume control of the robot at any time (essentially, after any discrete move or step).

A set of commands is given to an AI planning system, and it attempts to control the “robot” to achieve these goals. To do this, the planner maintains three (3) goal stacks: goals to do (TO-DO), goals done (DONE), and goals the planner needs to re-establish (RE-DO) because the human broke them after they were achieved. The planner does not sort the TO-DO or RE-DO goal stacks, but attempts to complete them in order from the top goal in the stack to the bottom. Thus, we assume that a goal ordering has already been achieved earlier in the planning cycle, and that we are now at execution time.

The AI system is designed to assume that a human is better able to determine how to deal with difficulties in the world. Thus, If the current goal cannot be achieved, the planner prompts

0123456789	Goals:
0 -----	TO-DO: ((BLOCK1 ((0 2)) AT))
1 -23-----	RE-DO: NIL
2 -13R-----	DONE: NIL
3 -43-----	
4 -----	
5 -----	
6 -----	
7 -----	
8 -----	
9 -----	

: AI system cannot achieve move of BLOCK1 to (0 2). Please assume
: control.

Figure 1: AI system requests human intervention.

the human for help to achieve it. Figure 1 demonstrates a case in which the AI system can accomplish a goal with the human's assistance that it could not achieve on its own. A deficiency of the planner is that it does not know it can push a block into its goal position by pushing another block into it. When it recognizes its goal is unachievable, it prompts the human for intervention. The human may then evaluate whether the goal should be modified or assume control of the system to help the planner achieve it. In this example, the human can just push BLOCK3 left or move it out of the way. With the human's assistance, complicated goals may be reduced to simpler goals achievable by the planner.

A more interesting behavior can be observed when an interaction occurs between the goals achieved by the planning system and the actions required by the human. In this case, the robot warns the human about the potential interaction, and asks for confirmation. Thus, the human is able to avoid accidentally interfering with the planning system. If another move would solve the same problem, the human might choose to avoid the interaction.

In Figure 2, we see a situation where trying to move the robot to the right will result in the human breaking a achieved goal namely the origin of BLOCK1 at position (7 7). The AI planner, monitoring the human's actions, will require confirmation that this is an intentional consequence. In this case, the human will avoid breaking the goal. Checking the goal stacks, one sees that no change has occurred.

If the human does decide to proceed with the move, the planner is able to recognize that a goal has been made invalid (i.e. "broken"). This goal is moved to a RE-DO stack, and will be re-established when the planner regains control. In this manner, the planning system is able to avoid taking later steps which rely on previously achieved goals which are no longer valid.

```

0123456789
0 -----
1 -----
2 -----
3 -----
4 -----
5 -----
6 -----
7 -----R11-
8 --2-----333
9 -----

```

```

Current Goals:    TO-DO: ((BLOCK2 ((7 6)) AT) (BLOCK3 ((7 8)) AT))
                  RE-DO: NIL
                  DONE: ((BLOCK1 ((7 7)) AT))
Human Action:     move right
System Response:  Moving to square (5 4) will break an achieved goal.
                  Are you sure? (Y/[N])
Human Action:     N
Current Goals:    TO-DO: ((BLOCK2 ((7 6)) AT) (BLOCK3 ((7 8)) AT))
                  RE-DO: NIL
                  DONE: ((BLOCK1 ((7 7)) AT))

```

Figure 2: AI makes human aware of interaction

In Figure 3, we see that the human is not directly pushing a block from a required location, but as a consequence of the command to push BLOCK3 the already completed goal involving BLOCK1 will be broken. Prompted again for confirmation, the operator this time confirms the action. At this point we see that in the planners goal tables, the BLOCK1 goal has been moved from the DONE stack to the RE-DO stack. Because the AI system monitored the human's actions, it realizes that goal needs to be re-achieved, and plans accordingly. (Note that the goal involving BLOCK3, which was at its target destination, was not moved because the planner only sees the top goal in the stack, and had not therefore achieved that goal.)

In addition, beneficial actions may also be observed. This, if the AI system recognizes that the human has moved the robot such that the planner's current goal is achieved, the system is aware of this and moves the goal to the DONE stack. Had the human not re-achieved the goal, that would have become the planner's current goal.

Finally, to remphasize, at each stage in the interaction the AI planning system is maintaining an accurate world model. goal. Because it has been monitoring the human's actions, it knows where the robot is, where each block is, and the correct contents of the goal stacks. Thus, upon

```

0123456789
0 -----
1 -----
2 -----
3 -----
4 -----
5 -----
6 -----
7 -----11-
8 --2----333
9 -----R--

```

```

Current Goals:   TO-DO: ((BLOCK2 ((7 6)) AT) (BLOCK3 ((7 8)) AT))
                  RE-DO: NIL
                  DONE:  ((BLOCK1 ((7 7)) AT))
Human Action:    move up
System Response: Moving to square (5 4) will break an achieved goal.
                  Are you sure? (Y/[N])
Human Action:    Y
Current Goals:   TO-DO: ((BLOCK2 ((7 6)) AT) (BLOCK3 ((7 8)) AT))
                  RE-DO: ((BLOCK1 ((7 7)) AT))
                  DONE:  NIL

```

Figure 3: Goal moved to RE-DO stack

regaining control it plans its current goal without having to reestablish the positions of the blocks.

4 Real-time Considerations

While the simple system we have implemented shows the need for the AI system to monitor human operation, in the real world the problem is obviously significantly more complex. The most significant difference is that the human control is *not* performed by giving easily recognizable commands to an AI system, but rather by the continuous operation of the telerobotic device. Thus, even the simple recognition of the human's invalidating of a previously achieved goal becomes a major problem.

For monitoring to be applicable in the real world, as opposed to a simulation environment, a large amount of run-time support for real-time computing is necessary. Our current research (Hendler, 1989) focuses on examining shared resource control in a real-time environment via the use of MARUTI, a hard real-time operating system developed by the Systems Research Group

at the university of Maryland (Levi et. al, 1988). The MARUTI OS is designed to support real-time applications on a variety of hardware platforms. MARUTI supports guaranteed-service scheduling, in which jobs that are accepted by the system are verified to satisfy general time constraints. In MARUTI, the system automatically verifies the schedulability of each component of a job with respects to its constraints and those of other jobs in the system.

The use of real-time in a telerobotic environment also poses a problem. The tolerance for response times must be large enough that the human may effectively interact, and without obsolete information. However, when there is a time lag, as in space applications, having the AI system monitor the human actions may be especially useful as the human might not otherwise get feedback until damage is already done. One does not, for example, wish an operator to get a warning about a potential interaction after it has occurred.

5 Conclusions

In this paper we have discussed the need for having an AI system monitor human usage of a shared resource. As discussed, we have demonstrated that this allows greater efficiency for the AI system, as an accurate world model can be maintained, and also allows for early detection of interactions enabling the discovery of cooperation or the avoidance of conflicts.

6 References

Hendler, J.A. "Real Time Planning" *Proceedings AAAI Spring Symposium on Planning and Search*, March, 1989 (In Press).

Levi, S., Tripathi, S., Carson, S. and Agrawala, A *The MARUTI Hard Real-Time Operating System* Technical Report, Computer Science Dept., University of Maryland, 1988.

Sanborn, J. and Hendler, J. "Monitoring and Reacting: Planning in dynamic domains" *International Journal of AI and Engineering*, Volume 3(2), April, 1988. p.95.